

数値計算方法、数値計算装置および数値計算用プログラム

発明の背景

技術分野

本発明は、コンピュータを用いた数値計算技術に関するものである。

関連技術の説明

従来から、圧力、温度などの物理量を対象とした制御技術において、逐次近似解法アルゴリズムの利用がなされている。そして、制御対象の各パラメータの解を、高速かつ高精度に求めることができる収束性の高い逐次近似解法アルゴリズムとして、残差切除法が知られている。さらに、特開2001-134304号公報では、残差切除法において収束性の更なる向上を実現するような技術が、すでに提案されている。

残差切除法のような逐次近似解法アルゴリズムでは、近似解に修正量を加えていくことによって解の精度を高めているが、修正量の計算には過去の修正量の列が用いられ、この列が長くなればなるほど修正量の精度が向上する。ところがその一方で、過去の修正量の列を長くすればとるほど、計算に必要な記憶領域は増大し、かつ、修正量を求めるための演算時間も長くなる、という問題が生じる。

発明の簡単な要旨

本発明は、逐次近似解法アルゴリズムを用いた数値計算において、計算精度を保ちつつ、計算に必要な記憶領域を抑えるとともに、計算時間を短縮することを目的とする。

具体的には、本発明は、物理量 U が満たすべき偏微分方程式を離散化したときの係数行列 (N 行 N 列、 N は正の整数) を A 、非斉次項 (ソース項) を f としたとき、コンピュータを用いて、 $A \cdot U = f$ を解き、物理量 U の数値計算を行うための方法、装置、またはプログラムを記録した記録媒体であって、

物理量 U の初期値 U^0 を設定し、繰り返し回数 m に初期値として 0 を、摂動量 ϕ の初期値として 0 を与え、残差 r の初期値 r^0 として $(f - A \cdot U^0)$ を設定し、繰り返し回数 m をインクリメントしながら、 $A \cdot \phi = r^m$ の予測近似値 Ψ^m を、内

部ソルバを有する第1演算部によって反復計算によって求める第1のステップと

残差 r^m の L_2 ノルムを最小とする修正近似値 ϕ^m を、第2演算部によって最適化ルーチンによって、予測近似値 Ψ^m から求め、近似解 U^{m+1} として $(U^m + \phi^m)$ を与え、残差 r^{m+1} として $(r^m - A \cdot \phi^m)$ を与える第2のステップとを、近似解 U^m が収束するまで繰り返し実行するものであり、

前記第2のステップは、求めたベクトル列 $A \cdot \phi^m$ の要素を、所定のサンプリング方法によってサンプリングして、メモリに保存するものであり、かつ、修正近似値 ϕ^m を求めるために用いる残差最小化係数 $\alpha_{l,m}$ ($l = 1, \dots, L$) を、前記メモリに保存されているベクトル列 $A \cdot \phi^k$ ($k = m - L + 1, \dots, m - 1$) の要素を用いて近似的に求めるものである。

そして、前記第2のステップは、ベクトル列 $A \cdot \phi^m$ の要素 b_1, b_2, \dots, b_n のサンプリングにおいて、要素 b_i ($i \in \Omega$) を選択する(ただし、

$$\Omega = \{i : \text{mod}[i, lg] = 1\} \cup \{i : |f_i / a_{ii}| > \beta\}$$

lg は整数、 β は実数、 f_i はソース項 f の要素、 a_{ii} は行列 A の i 行 i 列の対角項) ものとするのが好ましい。

図面の簡単な説明

図1は残差切除法を用いた数値計算方法を示すフローチャートである。

図2は本実施形態に係るベクトル列要素のサンプリングを概念的に示す図である。

図3は本発明の評価結果を示す図である。

図4は数値計算装置の構成を概念的に示すブロック図である。

発明の詳細な説明

以下、本発明の一実施形態について、図面を参照して説明する。

図1は本発明の一実施形態に係る数値計算方法を示すフローチャートである。

また図4は本実施形態に係る数値計算装置1の構成を示すブロック図である。

(残差方程式の定式化)

いま求めたい物理量を U とし、その物理量 U が満たすべき偏微分方程式を離散化したときの係数行列 (N 行 N 列、 N は正の整数) を A 、非斉次項 (ソース項) を f とすると、解くべき式は、

$$A \cdot U = f \quad \cdots (1)$$

で表される。この式は、一般的には、SOR法、ADI法等の逐次近似法によって解くことができる。

本発明では、以下のような解法を用いる。

式 (1) の解 U^∞ を、近似解 U と摂動量 (真の解との差) ϕ とによって、次のように表す。

$$U^\infty = U + \phi \quad \cdots (2)$$

本発明に係る解法では、摂動量 ϕ を求めて近似解 U を修正していくことによって、式 (1) の真の解 U^∞ を求める。

ここで、近似解 U に対する残差 r を次のように定義する。

$$r = f - A \cdot U \quad \cdots (3)$$

式 (1) ~ (3) から、

$$A \cdot (U + \phi) = f$$

$$\begin{aligned} \therefore A\phi &= f - AU \\ &= r \end{aligned}$$

したがって、摂動量 ϕ を求めるためには、次の式を解けばよい。

$$A\phi = r \quad \cdots (4)$$

(残差切除法のアルゴリズム)

本発明に係るアルゴリズムでは、式 (4) の収束解を求めるのではなく、ADI法などによって最も収束勾配の急な最小単位の反復で予測近似値 Ψ を求め、求めた予測近似値 Ψ を、最適化制御ルーチンに供給する。そして、最適化制御ルーチンの実行結果が所定の条件を満たすまで、予測近似値 Ψ の算出を繰り返し実行する。

<最適化制御ルーチン>

いま、繰り返し回数を m としたとき、残差の L_2 ノルムを最小とする合成摂動量 ϕ^m と新しい近似解 U^{m+1} を次のように定義する。

$$\phi^m = \alpha_{1,m} \psi^m + \sum_{l=2}^L \alpha_{l,m} \phi^{m-l+1} \quad \dots (5)$$

$$U^{m+1} = U^m + \phi^m \quad \dots (6)$$

$\alpha_{l,m}$ ($l = 1, 2, 3, \dots, L$) は残差最小化係数であり、後述する計算方法によって求まる定数である。残差の L2 ノルムがより小さくなるように、次のように残差の最小化を行う。

新しい近似解 U^{m+1} が式 (6) で表されるとき、これに対する残差 r^{m+1} は、式 (3) から、次のように表される。

$$r^{m+1} = f - A \cdot U^{m+1} \quad \dots (7)$$

式 (7) に式 (6) を代入し、式 (3), (5) を用いると、

$$r^{m+1} = r^m - \alpha_{1,m} A \cdot \psi^m - \sum_{l=2}^L \alpha_{l,m} A \cdot \phi^{m-l+1} \quad \dots (8)$$

近似解 U^{m+1} に対する残差 r^{m+1} の L2 ノルム $\| r^{m+1} \|$ は、 $(r^{m+1})^2$ の全点の総和の平方根として、式 (9) で与えられる。

$$\begin{aligned} & \| r^{m+1} \| \\ &= \sqrt{ \sum_i^N (r^m - \alpha_{1,m} A \cdot \psi^m - \sum_{l=2}^L \alpha_{l,m} A \cdot \phi^{m-l+1})^2 } \quad \dots (9) \end{aligned}$$

この残差 r^{m+1} の L2 ノルム $\| r^{m+1} \|$ を最小にするように、式 (9) の残差最小化係数 $\alpha_{l,m}$ ($l = 1 \sim L$) を最小二乗法で定める。すなわち、

$$\begin{aligned} \partial / \partial \alpha_{l,m} (\| r^{m+1} \|^2) &= 0 \\ l &= 1, 2, 3, \dots, L \quad \dots (10) \end{aligned}$$

式 (10) は $\alpha_{l,m}$ の L 元連立方程式となり、これを数値的に解くことによって $\alpha_{l,m}$ が定まる。 $\alpha_{l,m}$ が定まると、式 (5) から ϕ^m が求まり、式 (6) から U^{m+1} が求まる。残差最小化係数 $\alpha_{l,m}$ の計算方法については後述する。

m をインクリメントしながら、同様のルーチンを繰り返すことによって、式 (9) の残差の L2 ノルムを零または最小にする解 U^m に収束する。

図1に示すフローチャートに従って、本実施形態に係る数値計算方法について説明する。この数値計算方法は、例えばCD-ROM2のような記録媒体に記録されている、当該方法を実現するためのプログラムを、コンピュータに実行させることによって、実施することができる。

まず、ステップS1において、対象物理量 U の初期値 U^0 を設定する。そして、ステップS2において、繰り返し回数 m に初期値として0を、摂動量 ϕ の初期値として0を与え、残差 r の初期値 r^0 として $(f - AU^0)$ を設定する。

以下、近似解 U^m が収束するまで、繰り返し回数 m をインクリメントしながら、ステップS3～S7を繰り返し実行する。

ステップS3～S6では、ADI法、SOR法、CG法などの逐次近似法を実行する内部ソルバ11を有する第1演算部10によって、

$$A \cdot \phi = r^m$$

の近似解（予測近似値） Ψ^m を求める。ところがこの式は、既存の逐次近似法で解こうとすると、実用問題に対しては収束までに多くの繰り返しを必要とし、必然的に計算機使用時間も膨大となる。そこで本実施形態では、逐次計算の反復回数の上限 N を設定する（ステップS5）。

ステップS3～S6において予測近似値 Ψ^0 が求められると、ステップS7において、この予測近似値 Ψ^0 を用いて、第2演算部20による最適化ルーチンによって、次回の残差 r^1 が最小となるように、修正近似値 ϕ^0 を求める。この修正近似値 ϕ^0 によって、第1次の物理量の近似値 U^1 および第1次の残差 r^1 は、式（2）、（3）から、次式で求められる。

$$U^1 = U^0 + \phi^0 \quad \dots (12)$$

$$r^1 = f - AU^1 \quad \dots (13)$$

式（13）に式（12）を代入して、

$$\begin{aligned} r^1 &= f - A(U^0 + \phi^0) \\ &= f - AU^0 - A\phi^0 \\ &= r^0 - A\phi^0 \quad \dots (14) \end{aligned}$$

よって、式（14）により、第1次の残差 r^1 が求められる。

ステップS9で m をインクリメントしてステップS3に戻り、同様の計算を行

う。このような処理を繰り返すことによって、

$$(U^2, r^2), (U^3, r^3) \dots (U^m, r^m) \dots$$

が順次求められ、残差ノルム $\|r^m\|$ が減少していく。

繰り返し回数が m のとき、ステップ S 4 において予測近似値 ψ^m を求める方程式は、

$$A\phi = r^m$$

となり、ステップ S 7 では、最適化制御ルーチンによって最適化された修正近似値 ϕ^m によって、

$$U^{m+1} = U^m + \phi^m$$

$$r^{m+1} = r^m - A\phi^m$$

と求められる。

(残差最小化係数 α_l の求め方)

いま、表記を簡単にするために、次のように δ を導入する。

$$\delta^m = \psi^m$$

$$\delta^{m-l+1} = \phi^{m-l+1} \quad (l = 2, 3, \dots, L)$$

とおくと、修正近似値 ϕ^m は、

$$\phi^m = \sum_{l=1}^L \alpha_{l,m} \delta^{m-l+1}$$

と表される。よって、

$$\begin{aligned} r^{m+1} &= r^m - A\phi^m \\ &= r^m - A \sum_{l=1}^L \alpha_{l,m} \delta^{m-l+1} \\ &= r^m - \sum_{l=1}^L \alpha_{l,m} A \delta^{m-l+1} \end{aligned}$$

Σ 記号を外して表記すると、

$$r^{m+1} = r^m - (\alpha_{1,m} A \delta^m + \alpha_{2,m} A \delta^{m-1} + \alpha_{3,m} A \delta^{m-2} + \dots + \alpha_{L,m} A \delta^{m-L+1})$$

となる。

残差最小化係数 α_m は、例えば最小自乗法を用いて、 $(m+1)$ 次の残差ノルム $\|r^{m+1}\|$ (自乗和の平方根) を最小にするという条件を与えることによって、求められる。

$$\begin{aligned}\|r^m\| &= \sqrt{\sum_{i=1}^N (r^{m+1})^2} \\ \|r^m\|^2 &= \sum_{i=1}^N (r^{m+1})^2 \\ &= \sum_{i=1}^N \left(r^m - \sum_{l=1}^L \alpha_l A \delta^{m-l+1} \right)^2\end{aligned}$$

これを α_m で微分すると、

$$\partial / \partial \alpha_m (\|r^{m+1}\|^2) = 0$$

この式を α_m について解くことによって、残差最小化係数 α_m を求めることができる。微分を実行すれば、

$$\begin{aligned}\partial / \partial \alpha_m (\|r^{m+1}\|^2) &= 2 \sum_{i=1}^N \left[\left(r^m - \sum_{l=1}^L \alpha_l A \delta^{m-l+1} \right) \cdot (-A \delta^{m-l'+1}) \right] \\ \therefore \sum_{i=1}^N (A \delta^{m-l'+1} \sum_{l=1}^L \alpha_l A \delta^{m-l+1}) &= \sum_{i=1}^N (r^m A \delta^{m-l'+1}) \\ (\delta^m &= \Psi^m \quad \delta^{m-l'+1} = \phi^{m-l'+1} \quad (l=2, 3, \dots, L))\end{aligned}$$

となる。ここで、 $l'=1, 2, \dots, L$ である。この連立一次方程式を解くことによって、残差最小化係数 α_m を求めることができる。

例えば、 $L=3$ のときは、上式は、

$$\begin{aligned}\alpha_1 \sum (A \Psi^m)^2 + \alpha_2 \sum (A \Psi^m A \phi^{m-1}) + \alpha_3 \sum (A \Psi^m A \phi^{m-2}) &= \sum (r^m A \Psi^m) \\ \alpha_1 \sum (A \phi^{m-1} A \Psi^m) + \alpha_2 \sum (A \phi^{m-1})^2 + \alpha_3 \sum (A \phi^{m-1} A \phi^{m-2}) &= \sum (r^m A \phi^{m-1})\end{aligned}$$

$$\alpha_1^m \Sigma (A \phi^{m-2} A \Psi^m) + \alpha_2^m \Sigma (A \phi^{m-2} A \phi^{m-1}) + \alpha_3^m \Sigma (A \phi^{m-2})^2 = \Sigma (r^m A \phi^{m-2})$$

のような3元一次連立方程式になる。そして、

$$\phi^m = \alpha_1^m \Psi^m + \alpha_2^m \phi^{m-1} + \alpha_3^m \phi^{m-2}$$

$$r^{m+1} = r^m - (\alpha_1^m A \Psi^m + \alpha_2^m A \phi^{m-1} + \alpha_3^m A \phi^{m-2})$$

$$= r^m - A \phi^m$$

$$U^{m+1} = U^m + \phi^m$$

となる。

<ベクトル列のサンプリング>

すでに述べたとおり、残差最小化係数 α_i^m を求めるためには、上で示したような連立一次方程式を解く必要がある。ところで、上式では、 $A \cdot \phi^{m-1}$ 、 $A \cdot \phi^{m-2}$ 、…といったベクトル列が頻繁に現れる。このため、本実施形態では、ステップS7において、求めたベクトル列 $A \cdot \phi^m$ の要素をメモリ30に保存しておくことにする。これにより、修正近似値 ϕ^m を求めるために用いる残差最小化係数 α_i^m ($i=1, \dots, L$)を、すでにメモリ30に保存されているベクトル列 $A \cdot \phi^k$ ($k=m-L+1, \dots, m-1$)の要素を用いて求めることができるので、計算効率を向上させることができる。

さらに、本実施形態では、ベクトル列 $A \cdot \phi^m$ の保存方法を工夫することによって、メモリ容量と、計算時間とを節約する方法を提案する。すなわち、ベクトル列 $A \cdot \phi^m$ の全ての要素 b_1, b_2, \dots, b_N をメモリ30に保存するのではなく、一部の要素をサンプリングして保存するようにする。

ここで、集合 $\{1, 2, \dots, N\}$ の部分集合 Ω を、次のように定義する。

$$\Omega = \{i : \text{mod}[i, lg] = 1\} \cup \{i : |f_i / a_{ii}| > \beta\}$$

lg は整数パラメータであり、 β は実数パラメータである。 a_{ii} は行列 A の i 行 i 列の要素すなわち対角項、 f_i はソース項 f の要素である。 $lg=1$ のとき、または、 $\beta=0$ のときには、 $\Omega \equiv \{1, 2, \dots, N\}$ となる。そして、ベクトル列 $A \cdot \phi^m$ の要素を保存する際には、全 N 要素 b_1, b_2, \dots, b_N を保存するのではなく、行番号 i が部分集合 Ω に含まれる要素のみ、すなわち要素 b_i ($i \in \Omega$)のみを選択して保存する。言い換えると、上のような内積計算を、サンプリング要素のみの和によって近似して行う。

図2はベクトル列 $A \cdot \phi^m$ の要素のサンプリング例を概念的に示す図である。

同図中、(a)はソース項 f の各要素 f_i の分布を示すグラフであり、行列 A の対角項 $a_{i,i}$ によって正規化している。(b)は(a)の分布に対するサンプリング例であり、例Aは全要素を選択する場合、例Bは所定間隔おき(5要素おき)に選択する場合、そして例Cは部分集合 Ω に属する要素を選択する本実施形態の場合である。ここでは、実数パラメータ β の値は、 $|f_i / a_{i,i}|$ の最大値を基準にして設定しており、 $\beta = 0.98 \max(|f_i / a_{i,i}|)$ としている。

図2を見ると分かるように、本実施形態のように部分集合 Ω に属する要素を選択した場合(例C)では、空間的に均等なサンプリング(Δ 印)と、物理現象の特徴を踏まえたいわば局所的なサンプリング(\circ 印と \odot 印)とが、組み合わせられている。なお、もちろん、サンプリングの手法はここで述べたものに限られるものではなく、例Bのような空間的なサンプリングのみを行ってもよいし、物理現象を踏まえた局所的なサンプリングのみを行ってもよい。例えば、部分集合 Ω を次のように定めてもよい。

$$\Omega = \{i : \text{mod}[i, lg] = 1\}$$

$$\Omega = \{i : |f_i / a_{i,i}| > \beta\}$$

ただし、計算精度を保ちつつ、メモリ容量の節約や計算時間の短縮を実現するためには、空間的なサンプリングと、物理現象を踏まえた局所的なサンプリングとを組み合わせる行うのが好ましい。

<本発明の評価>

本発明の効果を評価するために、実際にコンピュータを用いて、共通の式の数値計算を、残差切除法を用いて、サンプリングを行わない場合(CASE1)と、上述の部分集合 Ω を用いたサンプリングを用いた場合(CASE2)とで、実行した。対象とした式は、ポアソン方程式を三次元構造格子上で二次精度差分を用いて離散化した連立一次方程式であり、内部ソルバ11にはSOR法を用い、残差最小化係数 α_m の個数は15個($L=15$)とした。また、 $lg=7$ 、 $\beta = 0.98 \max(|f_i / a_{i,i}|)$ とした。

図3は評価結果を示している。図3(b)において、縦軸は相対残差 $\|r\| / \|f\|$ であり、横軸はCPU時間である。図3から、本実施形態に係るサンプリ

ングを用いた場合には、サンプリングを行わない場合に比べて、計算時間は約 15%短縮されており、使用されるメモリ容量が約 25%少なくなっていることが分かる。

なお、ここで説明した数値計算方法は、様々な分野に利用可能である。例えば、流体解析のような解析技術に利用できることはいうまでもなく、車両の走行制御のような制御技術にも利用できる。

以上のように本発明によると、修正近似値 ϕ^m を求めるために用いる残差最小化係数 α_l^m ($l = 1, \dots, L$) が、メモリに保存されているベクトル列 $A \cdot \phi^k$ ($k = m - L + 1, \dots, m - 1$) の要素を用いて近似的に求められるので、計算効率を向上させることができる。しかも、ベクトル列 $A \cdot \phi^m$ の要素がサンプリングされてメモリに保存されるので、計算に必要なメモリ容量を抑えるとともに、計算時間を短縮することができる。

クレーム

1. 物理量 U が満たすべき偏微分方程式を離散化したときの係数行列 (N 行 N 列、 N は正の整数) を A 、非斉次項 (ソース項) を f としたとき、コンピュータを用いて、

$$A \cdot U = f$$

を解き、物理量 U の数値計算を行う方法は、

物理量 U の初期値 U^0 を設定し、

繰り返し回数 m に初期値として 0 を、摂動量 ϕ の初期値として 0 を与え、残差 r の初期値 r^0 として $(f - A \cdot U^0)$ を設定し、

繰り返し回数 m をインクリメントしながら、

$A \cdot \phi = r^m$ の予測近似値 ψ^m を、内部ソルバを有する第 1 演算部によって、反復計算によって求める第 1 のステップと、

残差 r^m の L_2 ノルムを最小とする修正近似値 ϕ^m を、第 2 演算部によって、最適化ルーチンによって、予測近似値 ψ^m から求め、近似解 U^{m+1} として $(U^m + \phi^m)$ を与え、残差 r^{m+1} として $(r^m - A \cdot \phi^m)$ を与える第 2 のステップとを、近似解 U^m が収束するまで、繰り返し実行するものであり、

前記第 2 のステップは、

求めたベクトル列 $A \cdot \phi^m$ の要素を、所定のサンプリング方法によってサンプリングして、メモリに保存するものであり、かつ、

修正近似値 ϕ^m を求めるために用いる残差最小化係数 $\alpha_{i,m}$ ($i = 1, \dots, L$) を、前記メモリに保存されているベクトル列 $A \cdot \phi^k$ ($k = m - L + 1, \dots, m - 1$) の要素を用いて、近似的に求める。

2. クレーム 1 において、

前記第 2 のステップは、

ベクトル列 $A \cdot \phi^m$ の要素 b_1, b_2, \dots, b_N のサンプリングにおいて、要素 b_i ($i \in \Omega$) を選択する (ただし、

$$\Omega = \{i : \text{mod}[i, l_g] = 1\} \cup \{i : |f_i / a_{i,i}| > \beta\}$$

l_g は整数、 β は実数、 f_i はソース項 f の要素、 $a_{i,i}$ は行列 A の i 行 i 列の対

角項)。

3. 物理量 U が満たすべき偏微分方程式を離散化したときの係数行列 (N 行 N 列、 N は正の整数) を A 、非斉次項 (ソース項) を f としたとき、

$$A \cdot U = f$$

を解き、物理量 U の数値計算を行う装置は、

物理量 U の初期値 U^0 を設定し、

繰り返し回数 m に初期値として 0 を、摂動量 ϕ の初期値として 0 を与え、残差 r の初期値 r^0 として $(f - A \cdot U^0)$ を設定し、

繰り返し回数 m をインクリメントしながら、

$A \cdot \phi = r^m$ の予測近似値 ψ^m を、内部ソルバを有する第 1 演算部によって、反復計算によって求める第 1 のステップと、

残差 r^m の L_2 ノルムを最小とする修正近似値 ϕ^m を、第 2 演算部によって、最適化ルーチンによって、予測近似値 ψ^m から求め、近似解 U^{m+1} として $(U^m + \phi^m)$ を与え、残差 r^{m+1} として $(r^m - A \cdot \phi^m)$ を与える第 2 のステップとを、近似解 U^m が収束するまで、繰り返し実行するものであり、

前記第 2 のステップは、

求めたベクトル列 $A \cdot \phi^m$ の要素を、所定のサンプリング方法によってサンプリングして、メモリに保存するものであり、かつ、

修正近似値 ϕ^m を求めるために用いる残差最小化係数 $\alpha_{i,m}$ ($i = 1, \dots, L$) を、前記メモリに保存されているベクトル列 $A \cdot \phi^k$ ($k = m - L + 1, \dots, m - 1$) の要素を用いて、近似的に求める。

4. クレーム 3 において、

前記第 2 のステップは、

ベクトル列 $A \cdot \phi^m$ の要素 b_1, b_2, \dots, b_N のサンプリングにおいて、要素 b_i ($i \in \Omega$) を選択する (ただし、

$$\Omega = \{i : \text{mod}[i, lg] = 1\} \cup \{i : |f_i / a_{ii}| > \beta\}$$

lg は整数、 β は実数、 f_i はソース項 f の要素、 a_{ii} は行列 A の i 行 i 列の対

角項)。

5. コンピュータに、物理量 U が満たすべき偏微分方程式を離散化したときの係数行列 (N 行 N 列、 N は正の整数) を A 、非斉次項 (ソース項) を f としたとき、

$$A \cdot U = f$$

を解かせ、物理量 U の数値計算を行う数値計算用プログラムを記録した記録媒体であって、この数値計算用プログラムは、

物理量 U の初期値 U^0 を設定し、

繰り返し回数 m に初期値として 0 を、摂動量 ϕ の初期値として 0 を与え、残差 r の初期値 r^0 として $(f - A \cdot U^0)$ を設定し、

繰り返し回数 m をインクリメントしながら、

$A \cdot \phi = r^m$ の予測近似値 Ψ^m を、内部ソルバを有する第 1 演算部によって、反復計算によって求める第 1 のステップと、

残差 r^m の L_2 ノルムを最小とする修正近似値 ϕ^m を、第 2 演算部によって、最適化ルーチンによって、予測近似値 Ψ^m から求め、近似解 U^{m+1} として $(U^m + \phi^m)$ を与え、残差 r^{m+1} として $(r^m - A \cdot \phi^m)$ を与える第 2 のステップとを、近似解 U^m が収束するまで、繰り返し実行するものであり、

前記第 2 のステップは、

求めたベクトル列 $A \cdot \phi^m$ の要素を、所定のサンプリング方法によってサンプリングして、メモリに保存するものであり、かつ、

修正近似値 ϕ^m を求めるために用いる残差最小化係数 $\alpha_{i,m}$ ($i = 1, \dots, L$) を、前記メモリに保存されているベクトル列 $A \cdot \phi^k$ ($k = m - L + 1, \dots, m - 1$) の要素を用いて、近似的に求める処理をコンピュータに実行させる。

6. クレーム 5 において、

前記第 2 のステップは、

ベクトル列 $A \cdot \phi^m$ の要素 b_1, b_2, \dots, b_N のサンプリングにおいて、要

素 b_i ($i \in \Omega$) を選択する (ただし、

$$\Omega = \{i : \text{mod}[i, l_g] = 1\} \cup \{i : |f_i| / a_{ii} > \beta\}$$

l_g は整数、 β は実数、 f_i はソース項 f の要素、 a_{ii} は行列 A の i 行 i 列の対角項)。

要約

ベクトル列 $A \cdot \phi^m$ の要素を、サンプリングしてメモリに保存する。このとき、空間的なサンプリングと、物理現象を踏まえた局所的なサンプリングとを組み合わせる。そして、修正近似値 ϕ^m を求めるために用いる残差最小化係数 α_l^m ($l = 1, \dots, L$) を、メモリに保存した $A \cdot \phi^k$ ($k = m - L + 1, \dots, m - 1$) の要素を用いて、近似的に求める。